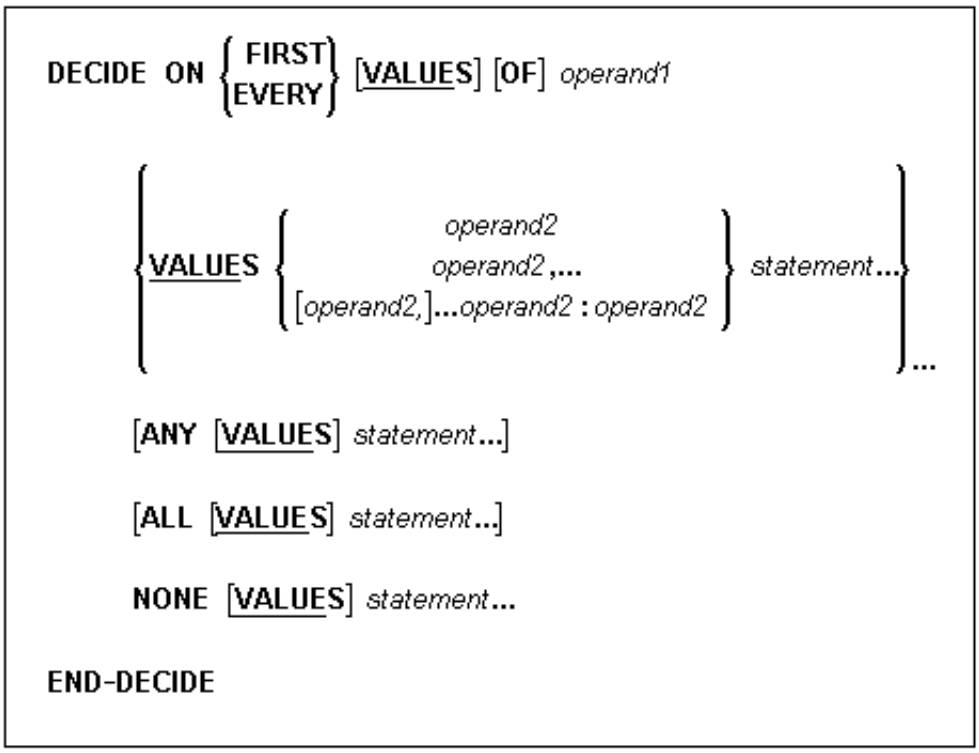
DECIDE ON



Operand	Possible Structure					Possible Formats												Referencing Permitted	Dynamic Definition
Operand1		S	A		N	A	N	P	I	F	B	D	T	L		G	O	yes	no
Operand2	C	S	A			A	N	P	I	F	B	D	T	L		G	O	yes	no

Related Statements: DECIDE FOR | IF

Function

The DECIDE ON statement is used to specify multiple actions to be performed depending on the value (or values) contained in a variable.

Note:
If **no** action is to be performed under a certain condition, you specify the statement IGNORE in the corresponding clause of the DECIDE ON statement.

FIRST/EVERY

With one of these keywords, you indicate whether only the first or every value that is found is to be processed.

Selection Field - operand1

As *operand1* you specify the field whose contents is to be checked.

VALUES Clause

With this clause, you specify the value *(operand2)* of the selection field, as well as the *statement(s)* which are to be executed if the field contains that value.

You can specify one value, multiple values, or a range of values optionally preceded by one or more values.

Multiple values must be separated from one another either by the input delimiter character (as specified with the session parameter ID) or by a comma. A comma must not be used for this purpose, however, if the comma is defined as decimal character (with the session parameter DC).

For a range of values, you specify the starting value and ending value of the range, separated from each other by a colon.

ANY

With ANY, you specify the *statement(s)* which are to be executed if *any* of the values in the VALUE clause are found. These statements are to be executed *in addition to* the statement specified in the VALUE clause.

ALL

With ALL, you specify the *statement(s)* which are to be executed if *all* of the values in the VALUE clause are found. These statements are to be executed *in addition* to the statement specified in the VALUE clause.

The ALL clause applies only if the keyword EVERY is specified.

NONE

With NONE, you specify the *statement(s)* which are to be executed if *none* of the specified values are found.

Example 1


```
/* EXAMPLE 'DECEX2': DECIDE ON (FIRST OPTION)
/************************************************************************
SET KEY ALL
INPUT 'TO UPDATE A RECORD, USE PF1 KEY' /
      'TO ADD    A RECORD, USE PF2 KEY' /
/************************************************************************
/* ROUTINE-UPD IS TO BE EXECUTED IF PF1 IS USED,
/* ROUTINE-ADD IS TO BE EXECUTED IF PF2 IS USED,
/* IF EITHER PF1 OR PF2 USED, END TRANSACTION IS TO BE EXECUTED,
/* IF NEITHER PF1 NOR PF2 ARE USED, NO STATEMENTS ARE TO BE EXECUTED.
/************************************************************************
DECIDE ON FIRST VALUE OF *PF-KEY
   VALUE 'PF1'
     PERFORM ROUTINE-UPD
   VALUE 'PF2'
     PERFORM ROUTINE-ADD
   ANY VALUE
     END TRANSACTION
     WRITE 'RECORD HAS BEEN MODIFIED'
   NONE VALUE
     IGNORE
END-DECIDE
/************************************************************************
END
```

Example 2

```
/* EXAMPLE 'DECEX2E': DECIDE ON (EVERY OPTION)
/* THIS EXAMPLE SHOWS THE EFFECT OF USING THE EVERY CLAUSE
/*****************************************************************
INPUT 'ENTER ANY VALUE BETWEEN 1 AND 9:' FIELD1(N1) (SG=OFF)
DECIDE ON EVERY VALUE OF FIELD1
  VALUE 1 : 4
    WRITE 'CONTENT OF FIELD1 IS 1-4'
  VALUE 2 : 5
    WRITE 'CONTENT OF FIELD1 IS 2-5'
  ANY VALUE
    WRITE 'CONTENT OF FIELD1 IS 1-5'
  ALL VALUE
    WRITE 'CONTENT OF FIELD1 IS 2-4'
  NONE VALUE
    WRITE 'CONTENT OF FIELD1 IS NOT 1-5'
END-DECIDE
/*****************************************
END
```

```
ENTER ANY VALUE BETWEEN 1 AND 9: 4
```

```
Page      1                                                  94-06-16  12:47:24

  CONTENT OF FIELD1 IS 1-4
  CONTENT OF FIELD1 IS 2-5
  CONTENT OF FIELD1 IS 1-5
  CONTENT OF FIELD1 IS 2-4
```

